Revisiting Hierarchical Quorum Systems*

Nuno Preguiça, J. Legatheaux Martins
Departamento de Informática
Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa
Quinta da Torre, 2825-114 Monte da Caparica,Portugal
{nmp,jalm}@di.fct.unl.pt

Abstract

In distributed systems it is often necessary to provide coordination among the multiple concurrent processes. Quorum systems provide a decentralized approach to provide such coordination that is resilient to node and communication link failures. Quorum systems are highly available and may be used to balance the load among the elements of the system. In this paper, we propose a modification to the hierarchical grid quorum system that leads to a smaller quorum size, better availability and load. We also propose a new hierarchical quorum construction based on the organization of elements in a triangular shape that presents better average quorum size, availability and load than other highly-available systems with almost optimal load.

1. Introduction

In a distributed system it is often necessary to achieve coordination among multiple concurrent processes. In the past, numerous solutions have been proposed to this problem (for example, see [2] for a survey on mutual exclusion algorithms). Quorum systems have been used as a basic tool to provide such coordination in different situations. For example, quorum systems have been used in data replication protocols [6, 7], location management algorithms [17], masking Byzantine failures [12], etc.

A quorum system is defined over an universe of N different processes (usually located on N different nodes) that can communicate by message exchange. A quorum system is a collection of subsets of processes that satisfy the intersection property, i.e., every pair of subsets has a nonempty intersection. Each subset is called a quorum.

The outline of a protocol based on a quorum system is the following. In order to enter a critical section to execute some action, the user requests permission to a quorum of processes. If all the processes give him the requested permission, the user has a lock and he may enter the critical section. After leaving the critical section the user should release his lock (from the permission-granting quorum). The intersection property guarantees that no two users will be granted permission to enter the critical section simultaneously (as long as each process only grants permission to a user after the previous user has released his lock). Note that this simplified outline is prone to deadlocks and a more elaborate algorithm must be devised (e.g. [6]).

Quorum systems are attractive because they provide a decentralized approach that tolerates failures, i.e., operation is still possible even in the presence of server crashes and/or network partitions (as quantified by their availability [15]). For example, several proposed quorum systems (e.g. [5, 16, 9]) present very high availability that tends to 1 very quickly as more processes are added to the system. Moreover, quorum systems are also interesting for very large-scale systems, because it is possible to make the size of quorums increase much slower than the system size. For example, in the grid protocol [3] the quorum size is approximately $2\sqrt{n}$ (e.g., in a system with 100 nodes the quorum size is 20). Therefore, it is possible to provide very high availability with a reasonable communication cost. Additionally, quorum systems can be used to execute load-balancing [18]: as only a small fraction of servers receive each request, if the quorum selection strategy is properly chosen, each server will have to handle only a small fraction of requests.

In [9] the authors have proposed a quorum system that presents asymptotically optimal availability (the availability tends to 1 as more elements are added to the system). This system is based on a hierarchical grid organization, and quorums are obtained recursively in the defined organization. In this paper we propose a modification to the original system that decreases the average quorum size and improves availability. The load of the modified algorithm is also lower.

*This work was partially supported by FCT Foundation, project number 33924/99.

1063-6927/01 $10.00 © 2001 IEEE

We also propose a new quorum system based on the hierarchical organization of elements in a triangular shape. This new algorithms presents constant quorum size that is smaller than the average quorum size in highly available systems with $O(1/\sqrt{n})$ load. The load of the system - $\sqrt{2}/\sqrt{n}$ is almost optimal and it is better than that presented by previously proposed highly available quorum systems. The availability also proved to be better in our analysis.

The remainder of this paper is organized as follows: section 2 discusses related work; section 3 presents basic definitions and results already established about quorum systems; in section 4 we propose a modification to the hierarchical grid quorum system; in section 5 we present the new hierarchical triangle quorum system; in section 6 we analyze the new construction and section 7 concludes the paper with some final remarks.

2. Related work

The first protocols using quorum systems use voting to define the quorums (e.g [5]) - a quorum is any set of elements with a combined number of votes larger than half of the total number of votes in the system. When all elements have 1 vote we have the majority system. The majority system presents the best possible availability when the individual failure probability of each element (that is considered equal to all elements) is $p < 0.5$ [15], but it requires quorums of size $\frac{n+1}{2} = O(n)$. To reduce the size of the quorums, the hierarchical quorum system (HQS) [8] is based on a n-ary tree construction where elements are the leaves. A quorum is formed recursively from the root node, obtaining a quorum in a majority of sub-trees. The quorum size in this system is $O(n^{0.63})$. An alternative process that also uses a tree construction has been proposed in [1] - in this system the quorums have different sizes. These systems present good availability, but their load is worse than the best possible (e.g. $O(n^{-0.37})$ for the HQS against $O(1/\sqrt{n})$ for the best load-balancing systems [14]).

An alternative method to reduce the size of quorums have been proposed in [13] using finite projective planes - this method uses quorums of size $\sqrt{n}$. However, it is only known how to construct this system in a small number of situations. Alternative ways to easily produce quorums of size $O(\sqrt{n})$ have been proposed based on the organization of elements in grids [3] or triangles [11]. The availability of these systems is poor [15] - it asymptotically tends to 0 as more elements are added to the system. An alternative triangle-based quorum system [15] does not present such bad availability. However the failure probability does not vanish as more elements are added ($F_p > p^{\frac{1}{p}}$) [15]. A similar analysis can be applied to the diamond-based quorum system proposed in [4].

In [9] the authors have used a hierarchical organization to propose a system that presents asymptotically good availability (that tends to 1 as more elements are added) and almost optimal load ($2/\sqrt{n}$). In [14] the authors present several quorum systems that have near optimal load and high availability with $O(\sqrt{n})$ quorum sizes. In [16] the authors present the CWlog quorum system that has small quorums (of size $O(\lg n)$) and optimal availability and load among systems with such small quorum size.

Recently, quorum systems have been used to mask Byzantine failures [12]. As the proposed Byzantine quorum systems extend ideas already used in normal quorum systems, we believe that the ideas proposed in this paper can also be adapted and used in Byzantine quorum systems.

3. Preliminaries

In this section we present some basic definitions, terminology and results used later on (we follow closely [14]).

Definition 3.1 *A quorum system $\mathcal{S} = \{S_1, \ldots, S_m\}$ is a collection of subsets $S_i \subseteq \mathcal{U}$ of a finite universe $\mathcal{U}$ that satisfies the intersection property: $P \cap R \neq \emptyset, \forall P, R \in \mathcal{S}$. The subsets $S_i \in \mathcal{S}$ are called quorums. A coterie is a quorum system $\mathcal{S}$, such that there are no $P, R \in \mathcal{S}, P \subset R$.*

In the study of the quality of a quorum system it is usual to use three metrics: the quorum size, the failure probability and the load of the system. The first one measures the number of nodes that need to be contacted to form a quorum. The second one measures the probability that all quorums are unavailable, i.e., that the system is unusable. The third one measures the frequency of access of each element of the system, i.e., the percentage of requests it has to process.

In the study of system availability, we use a simple probabilistic model of the failures (as usual in quorum systems literature). We assume that the elements (processes) of the system only fail by crashing and that all the failures are transient. The crashes are independent and all processes have the same crash probability equal to p (we use q to denote the survival probability). The failure probability of a quorum system (also called crash probability) is defined as follows.

Definition 3.2 *For every quorum $S \in \mathcal{S}$ let ε_S be the event that S is hit, i.e., at least one element $i \in S$ has failed. The failure probability of a quorum system $\mathcal{S}$ is the probability that all quorum are hit, i.e., $F_p(\mathcal{S}) = \mathcal{P}_p(\cap_{R \in \mathcal{S}} \varepsilon_R)$.*

An alternative way to determine the failure probability of a quorum system is to use the transversals of $\mathcal{S}$.

Proposition 3.1 ([15]) *A set T is a size-i transversal of a quorum system $\mathcal{S}$ if $|T| = i$ and for every $R \in \mathcal{S}, T \cap R \neq \emptyset$. Let $a_i^{\mathcal{S}}$ be the number of size-i transversal of $\mathcal{S}$, the failure probability of $\mathcal{S}$ is $F_p(\mathcal{S}) = \sum_{i=0}^{n} a_i^{\mathcal{S}} p^i q^{n-i}$*

The next propositions establish the best possible failure probability that a quorum system can present (and which quorum systems present such values).

Proposition 3.2 ([15]) *When $0 \leq p < \frac{1}{2}$, the coterie that presents best failure probability is the majority quorum system. When $\frac{1}{2} < p \leq 1$, the coterie that presents best failure probability is the singleton quorum system.*

From these results follows that, when $p > 0.5$, it is impossible to improve the availability introducing new elements in the singleton quorum system. Due to this reason, in this paper we restrict the failure probability analysis to the cases where $p \leq 0.5$. The load is also an important measure because it estimates the quality of the quorum system for performing load-balancing. For example, if the load is 0.2 it means that the busiest process only receives 20% of all requests (if the optimal quorum selection strategy is used). Therefore, either the system is able to receive more requests or the processes are able to perform other unrelated tasks.

Definition 3.3 *Let $\mathcal{S} = \{S_1, \ldots, S_m\}$ be a quorum system. $w \in [0,1]^m$ is a strategy for $\mathcal{S}$ if it is a probability distribution over the quorums $S_i \in \mathcal{S}$, i.e., $\sum_{i=1}^{m} w_i = 1$.*

In other words, a strategy gives the probability that a quorum S_j will be picked when the system is accessed. A given strategy induces a probability that the element i is accessed, which we call load on i. The system load is the load of the busiest element induced by the best possible strategy.

Definition 3.4 *Let w be a strategy for a quorum system $\mathcal{S}$. For an element $i \in \mathcal{U}$, the load induced by w on i is $l_w(i) = \sum_{S_j \ni i} w_j$. The load induced by w on a quorum system $\mathcal{S}$ is $\mathcal{L}_w(\mathcal{S}) = \max_{i \in \mathcal{U}} l_w(i)$. The system load on a quorum system $\mathcal{S}$ is $\mathcal{L}(\mathcal{S}) = \min_w\{L_w(\mathcal{S})\}$ where the minimum is taken over all strategies w.*

From the above definition, it is obvious that it is important to determine not only the load of a quorum system, but also the optimal strategy that induces that load (or at least, a nearly optimal one). Several results are known about the load of a quorum system.

Proposition 3.3 ([14]) *Let $c(\mathcal{S})$ be the cardinality of the smallest quorum in $\mathcal{S}$ over an universe $\mathcal{U}$ of n elements. For every quorum system $\mathcal{S}$, $\mathcal{L}(\mathcal{S}) \geq \frac{c(\mathcal{S})}{n}$ and $\mathcal{L}(\mathcal{S}) \geq \frac{1}{c(\mathcal{S})}$. Therefore it is immediate to establish that $\mathcal{L}(\mathcal{S}) \geq \frac{1}{\sqrt{n}}$.*

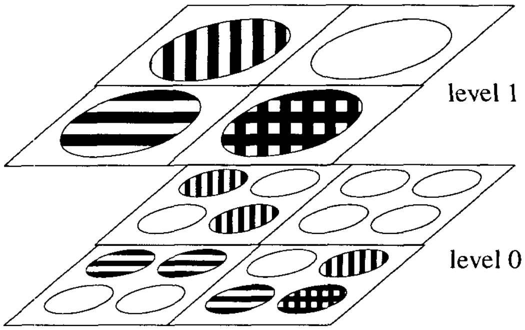

Figure 1. A 3-level hierarchical grid with 16 processes (level 2, not depicted, contains a single logical object). A read-write quorum is illustrated relying on row-covers (vertical lines) and full-lines (horizontal lines).

The previous results establish a lower bound on the load of any quorum system. Moreover, they also establish that this lower bound can only be reached in quorum systems with the smallest quorum size equal to $\sqrt{n}$.

4. Hierarchical T-Grid

In this section we will briefly present the hierarchical grid algorithm proposed in [9] and describe a small modification that improves the original algorithm in respect to failure probability, quorum size and load. We present some results that show this improvement.

4.1. Hierarchical grid [9]

A hierarchical grid (h-grid) organizes a number of processes into a multi-level hierarchy as follows. Processes are at level 0 of the hierarchy and logical objects are defined at higher levels. A logical object at level i ($i > 0$) is defined by a grid of $m_i \times n_i$ objects at level *i-1* (figure 1 depicts a two-level h-grid with 16 processes - note that nothing prevents two logical objects, in the same or in a different level, from being defined by grids of different sizes).

The h-grid protocol has been proposed to manage replicated data. Three operations are defined: read, blind write and read/write. Concurrent reads are allowed and concurrent blind writes are also allowed. However, concurrent reads and blind writes are not possible. Exclusive access to data is provided by the read/write operation. To coordinate concurrent accesses to replicated data, the authors propose the creation of three types of quorums to manage the respective operations: read quorums, write quorums and read/write quorums.

A **read quorum** is formed, as follows, obtaining a row-cover in the logical object on the top of the hierarchy. A row-cover in a level i object is formed (recursively) obtaining a row-cover in at least one object of every row of the corresponding level *i-1* grid. A row-cover in a level 0 object is defined as the self-object. In the example of figure 1, the top-leftmost element of level 1 has formed a row-cover using the top-leftmost and bottom-rightmost elements of the corresponding level 0 grid.

A **write quorum** is formed, as follows, obtaining a full-line in the logical object on the top of the hierarchy. A full-line in a level i object is formed (recursively) obtaining a full-line in all objects of at least one row of the corresponding level *i-1* grid. A full-line in a level 0 object is defined as the self-object. In the example of figure 1, the bottom-leftmost element of level 1 has formed a full-line using the elements in the top row of the corresponding level 0 grid.

A **read-write quorum** in a h-grid is formed through the union of a read and a write quorum. To guarantee the correctness of the defined operations, it is necessary to show that any pair of read and write quorums intersect (as a read-write quorum contains both a read and a write quorum, it will necessarily intersect any other read, write or read/write quorum). The read and the write quorums are obtained respectively by a row-cover and a full line. As the row-cover and the full-line are initially defined in the same logical object, it is obvious (from definitions) that in the next level a common object is used to obtain both the row-cover and the full-line. Using the same argument recursively, it is obvious that a common level 0 object integrates both sets (for a formal proof see [9]).

4.2. Hierarchical T-grid

The h-grid protocol (previously described) may be used to provide mutual exclusion using the read-write operation (and the correspondent read-write quorums - called simply as quorums where no confusion may arise). However, if mutual exclusion is the only operation necessary, the original algorithm imposes the integration of unnecessary elements in each quorum. From the informal correctness proof it is obvious that any two read-write quorums have an intersection of, at least, two elements - the read quorum of each read-write quorum intersects the write quorum of the other read-write quorum.

The hierarchical T-grid algorithm (h-T-grid) that we propose in this section removes unnecessary elements using an idea already proposed for the grid quorum system - a grid quorum can be obtained through the intersection of a full-line and one element from each row below the full line, thus removing the need to integrate a full row cover as proposed in [3]. It is easy to see that this new grid algorithm is correct because, for every pair of quorums, the partial row-cover of the quorum based on the higher full-line necessarily contains an element in the full-line used by the other quorum.

To define the h-T-grid quorum we start defining the global position of every level 0 object. In a hierarchical grid with every logical object defined as grids with the same dimensions, the global position is easily obtained organizing all level 0 objects in a large grid concatenating all grids of level 0 objects (as depicted in the level 0 of figure 1) - the global position of an element is its position in the grid (being (1,1) the position of the top-leftmost element). When grid dimensions are different it is necessary to guarantee that level 0 global positions reflect the relative positions of all parent logical objects.

Definition 4.1 *The global position of a level* 0 *object, obj, in a hierarchical grid with i levels is defined as a pair* $([x_{i-1}, x_{i-2}, \ldots, x_0], [y_{i-1}, y_{i-2}, \ldots, y_0])$ *where* (x_0, y_0) *is the position of obj in the level* 0 *grid where it is contained and* (x_n, y_n), $0 < n \leq i-1$, *is the position of the* n^{th} *parent of obj in the level* n *grid of logical objects where it is contained.*

Definition 4.2 *An object with global position* $([x^A_{i-1}, \ldots, x^A_0], [y^A_{i-1}, \ldots, y^A_0])$ *is above an object with global position* $([x^B_{i-1}, \ldots, x^B_0], [y^B_{i-1}, \ldots, y^B_0])$ *iff* $\exists\, 0 \leq j \leq i-1 : x^A_j > x^B_j \wedge (\forall\, n > j, x^A_n = x^B_n)$. *A level* 0 *object* X *is a topmost object in a set* S *of level* 0 *objects iff there is no object* $Y \in S : Y above X$ *(note that there may exist several topmost objects in a given set, but for any two topmost elements* T_1, T_2 *of a given set,* $\forall P, P above T_1 <=> P above T_2$*).*

A **partial row-cover** in respect to a given set S of level 0 objects is formed removing from a row-cover all objects that are above a topmost object of S. A **hierarchical T-grid quorum** is formed by the union of a full-line (defined as in the hierarchical grid) and a partial row-cover in respect to that full-line (both obtained from the logical object on the top of the hierarchy).

To prove the correctness of the h-T-grid quorums we must prove that a row-cover in respect to a given full-line has a non-empty intersection with any other full-line whose elements are not above a topmost object in the first full-line.

Theorem 4.1 *Given a full-line L, any partial row-cover in respect to L has a non-empty intersection with any other full-line M that has no element above any topmost element of L, i.e., for every topmost element of L, Q,* $\forall P \in M, \neg(P above Q)$.

Proof: Let's assume that that the partial row-cover has an empty intersection with the full-line M. We know that any full row-cover has a non-empty intersection with any full-line [9]. As a partial row-cover in respect to a set L is obtained from a row-cover removing all elements above a topmost of L, the intersection elements must be in the set of removed elements. However, we also know that the full-line M has no element above any topmost element of L thus leading to a contradiction. Therefore the partial row-cover should have a non-empty intersection with M. ■

Lemma 4.1 *Any two hierarchical T-grid quorums intersect.*

Proof: Let the first quorum be obtained as the full-line L and a partial row-cover relative to L, and the second quorum as the full-line M and a partial row-cover relative to M. Let's assume (without loss of generality) that the full-line M has no element above any topmost element of L. From theorem 4.1 it is known that the partial row-cover relative to L intersects the full-line M. Therefore, any two hierarchical T-grid quorums have a non-empty intersection. ■

It is interesting to note that any h-T-grid quorum still intersects with any full row-cover. Therefore it is still possible to manage replicated data using the read quorum defined in the h-grid and the quorum defined in the h-T-grid to manage the read and the exclusive write operations, respectively (if only these operations are necessary).

4.3. Analysis

Failure probability: In [9] the authors have already proved that the availability of the h-grid quorum system increases asymptotically as more levels are added to it (for all $p < p^* < 0.5$, with the actual value of p^* dependent on the dimensions of the sub-grids). It is obvious that the availability of the h-T-grid quorum system can not be worse than that of the h-grid. Therefore, it also increases asymptotically (and failure probability decreases, $F_p(\text{h-T-grid}) \to 0$).

To analyze the improvement of the h-T-grid over the traditional h-grid, we have determined the failure probability of several systems with different number of nodes. In table 1 we present the results obtained for systems with 9, 16 and 25 nodes organized in square grids (logical grids have size 2×2 whenever it is possible). From the results obtained it is possible to observe that in these configurations, the h-T-grid quorum system improves the failure probability in approximately $7.5\% - 10\%$. An interesting observation was that the improvement of the h-T-grid quorum system is much bigger when the number of lines is larger than the number of columns - for example, for a system with 24 nodes organized in a grid of 4 columns and 6 lines, the failure probability of the h-T-grid system is less than $1/3$ of the corresponding h-grid system and it is even better than the failure probability of the square grid with 25 nodes (without incurring in bigger quorum sizes). Moreover, it was possible to observe that although the h-T-grid presents an even bigger improvement from the results obtained in the h-grid, organizing the elements in a 3×8 grid leads to a worse failure probability than using the 4×6 grid. For systems with approximately 9 and 16 elements, similar results have been obtained. These results seem to indicate that the h-T-grid quorum system presents the best failure probability results with slightly rectangular grids (with more lines than columns) and that these results are much better than those presented by the best h-grid with similar number of elements.

Load and quorum size: In the h-grid quorum system, all quorums have the same size approximately equal to $2\sqrt{n} - 1$. Using the results of proposition 3.3 we can obtain that $\mathcal{L}(\text{h-grid}) \geq 2/\sqrt{n}$. As all quorums have the same size, it is obvious that each request induces in every element of the system a load approximately equal to $2\sqrt{n}/n = 2/\sqrt{n}$. If it is possible to determine a strategy that induces the same load in every element, $\mathcal{L}(\text{h-grid}) \approx 2/\sqrt{n}$. A simple strategy that achieves that property is to randomly select in each level the elements that are used to form the h-grid quorum, thus imposing equal responsibility to all elements.

In the h-T-grid quorum system the quorum size is variable - $\sqrt{n} \leq |\text{quorum}| \leq 2\sqrt{n} - 1$. The load induced in an element is the sum of the load induced when the element is part of a full-line and when the element is part of a partial row-cover. In the h-T-grid, the elements in the higher rows will be part of partial row-cover less frequently than those in the lower rows. Therefore, to distribute uniformly the load by all elements it is necessary to select more frequently quorums based on higher rows. Therefore, using such strategy, the average size of the selected quorums will be bigger that $\frac{\sqrt{n}+2\sqrt{n}-1}{2} \approx \frac{3}{2}\sqrt{n}$ and consequently the load induced will be greater than $\frac{\sqrt{n}+2\sqrt{n}-1}{2n} \approx \frac{3}{2\sqrt{n}}$.

The optimal strategy to minimize the load is to form quorums based on full-lines with all elements in the same line (and partial row-covers randomly selected). Then it is easy to calculate the probability that should be used to select each row as the base for a quorum - for example, for a square grid with 16 elements we would get an average quorum size of 5.8 elements and a load of 36.5% (against 5.5 and 34.375% from the lower bounds estimated before). However, this strategy does not use all quorums defined in the system. A simple modification to this strategy that uses all quorums is the following: when selecting the fragments that compose a full-line based on a given line, introduce a (small) probability to use elements from a lower line. Using such a strategy, we have obtained worse results, as expected - for the same square grid, the average quorum size was 5.9 and the load 41%. In real situations, the strategy to be used should be adapted taking into consideration the elements that are failed (as it should also be done in h-grid).

5. Hierarchical Triangle

In the hierarchical triangle quorum system (h-triang) the processes are organized in a triangle shape with i rows where the i^{th} row has i elements. This triangle is hierarchically organized in levels using the following recursive procedure. In the level m, a triangle composed of j rows is divided in two sub-triangles of level $m+1$ and a sub-grid. The sub-triangle 1 is composed by the top $\lfloor j/2 \rfloor$ rows of the level m triangle. The sub-grid is composed by the first $\lfloor j/2 \rfloor$ elements of rows $\lfloor j/2 \rfloor + 1$ to j. The sub-triangle 2 is composed by the remaining elements of the original level m triangle (forming a triangle with $j - \lfloor j/2 \rfloor$ rows). Triangles with a single line are not divided. In figure 2 we present the example of the logical division of a triangle with 5 rows. The original triangle that includes all the elements of the system defines a level 0 triangle.

With this organization, a **h-triang quorum is formed** obtaining a quorum in the triangle of level 0. A quorum in the triangle of level m is defined as follows:

- If the triangle has a single line, the quorum is composed by the element in the line.
- If the triangle has more than one line, a quorum can be obtained by the following three methods. Let T_1 and T_2 be the sub-triangles 1 and 2 of level $m+1$ and G be the sub-grid (all defined as explained before).
    1. If A is a quorum in T_1 and B is a quorum in T_2, $A \cup B$ is a quorum in the triangle of level m.
    2. If A is a quorum in T_1 and B is a row-cover in G (as defined in the h-grid - see section 4.1), $A \cup B$ is a quorum in the triangle of level m.
    3. If A is a quorum in T_2 and B is a full-line in G (as defined in the h-grid - see section 4.1), $A \cup B$ is a quorum in the triangle of level m.

To proof the correctness of the h-triang quorum system it is necessary to prove that any two quorums intersect.

Theorem 5.1 *Any two quorums defined in the hierarchical triangle quorum system intersect.*

Proof: (By induction on the number of lines) For a triangle with a single line, there is only one quorum that contains the element in the line. For a triangle of level m with $j > 1$ rows, several cases must be considered:

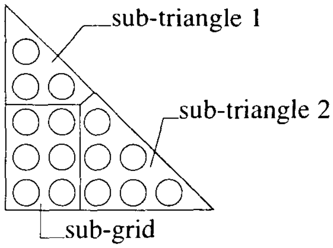

Figure 2. A triangle with 5 rows (15 processes) divided in two sub-triangles and a sub-grid (as defined in the hierarchical triangle quorum system).

- Two quorums defined using the same method obviously include quorums defined in the same sub-triangle of level $m+1$ with $i < j$ rows. By hypothesis, these quorums intersect.
- A quorum defined using method 1 and a quorum defined using methods 2 or 3 include a quorums defined in the same sub-triangle of level $m+1$ with $i < j$ rows. By hypothesis, these quorums intersect.
- A quorum defined using method 2 and a quorum defined using method 3 intersect in the sub-grid, because a full-line and a row-cover defined in a h-grid intersect [9].

Therefore, any two quorums defined in the hierarchical triangle quorum system intersect. ■

The availability of the proposed construction increases asymptotically as more levels are added to the system. Due to space limitations we will just sketch the proof. Analyzing the analytical function for the availability of a triangle it is easy to conclude that the availability increases if the availability of the sub-elements also increase. In [9] it has been proven that both the probability of getting a hierarchical row-cover and a hierarchical full-line increase asymptotically as more levels are added to the system. Using this information, it is easy to prove, by induction on the levels, that the availability of the h-triang also increase asymptotically.

Strategy that minimizes load: To minimize the load it is necessary to devise a strategy that induces an uniform load in all elements. To this end, it is necessary to take into consideration, in each triangle, the number of elements that compose the sub-triangles and the sub-grid and the number of elements necessary for the quorum in each one. A simple strategy that balances the load using all quorums is obtained selecting, at each level of the triangle, the different methods to form a quorum with probability w_1, w_2, and w_3 respectively, obtained solving the following equations (let c_1, c_2, c_3 be the number of elements in the sub-triangle 1, 2 and in the sub-grid; q_1, q_2 be the number of elements necessary to form a quorum in the sub-triangle 1 and 2; and q_{3l}, q_{3r} be the number of elements necessary to obtain a full-line and a row-cover in the sub-grid).

$$\begin{cases} w_1 + w_2 + w_3 = 1 \\ w_1 + w_2 = \frac{c_1}{q_1} k \\ w_1 + w_3 = \frac{c_2}{q_2} k \\ \frac{q_{3r}}{c_3} w_2 + \frac{q_{3l}}{c_3} w_3 = k \end{cases}$$

In the grids, full-lines (row-covers) are selected randomly, at each level, with probability proportional to the number of represented level 0 lines (columns). It is possible to verify, for a given system configuration, that this strategy uses each element with equal probability, thus inducing the best possible load.

Introducing new elements: The hierarchical nature of the h-triang construction makes it easy to introduce new elements in the structure improving the failure probability (without the need to introduce a new full-line). For example, it is possible to improve the availability of a level m triangle, replacing:

- A sub-triangle with n lines by one with $n + 1$ lines, in particular, a sub-triangle with 1 element by one with 3 elements (2 lines).
- Replacing a sub-grid with 1 element by a sub-grid with 2 elements (1 line and 2 columns).
- Replacing a sub-grid with $n \times n$ elements by a sub-grid with $(n + 1) \times (n + 1)$ elements.

As the availability of the level m triangle increases, it can be easily proved that the availability of the system also increases.

6. Analysis

In this section, we analyze the availability, quorum size and load presented by the proposed hierarchical triangle quorum system. We compare it to other quorum system constructions previously proposed in literature.

Failure probability: Due to the complication of determining the exact analytical expressions for the failure probability of quorum systems based on paths in graphs - Paths [14] and Y [10] -, our analysis is based on the enumeration of all possible configurations in systems with (almost) equal number of elements (thus, using the results of proposition 3.1). In table 2 and 3 we present the failure probability for different quorum systems with approximately 15 and 28 elements respectively (the results for the Y quorum system have been obtained in [10]). From the results presented it is possible to observe that the majority quorum system [5] and the hierarchical quorum system (HQS) [8] exhibit the lower failure probability. However, the size of the quorums used in these systems, $O(n)$ and $O(n^{0.63})$ respectively, is larger than the others.

From the systems that present $O(\sqrt{n})$ quorum sizes - h-T-grid, Paths (presented as the best construction proposed in [14]), Y and h-triang -, the last two exhibit the best results. It is worth to note that these two system organize the nodes in a triangle shape, while the other two use grid-based configurations. The CWlog system [16] uses quorums of variable sizes, some of which are smaller ($O(\lg n)$) than those used in the other systems. Nevertheless, this system also presents good failure probability for the analyzed number of nodes (and a failure probability that tends to 0 as more elements are added).

From the results obtained, it seems important to note that all systems present a very low failure probability even with a small number of nodes - e.g. in the h-triang with 15 elements, the failure probability is less than 0.07% when the individual failure probability of elements is 10%. In the studied configurations, the h-triang presents the best results.

Load and Quorum Size: As it has already been mentioned, the systems that present best availability - majority and HQS - use quorums larger than the others. From proposition 3.3, it follows that the load in these systems will also be larger than the load in system with $O(\sqrt{n})$ quorum sizes. In CWlog, although the smallest quorum has size $O(\lg n) < O(\sqrt{n})$, the biggest has size $O(n/\lg n) > O(\sqrt{n})$. The average size of the used quorums depends on the strategy used - for example, using the strategy proposed in [16] as a good tradeoff between quorum size and load, for a system with 14 (resp. 29) elements we have obtained an average quorum size of 4 (5.25) and a load of 55.5% (43.7%). From proposition 3.3 it follows than the small quorums lead to a load - $O(1/\lg n)$ - worse than the best possible - $O(1/\sqrt{n})$.

All the other analyzed systems present $O(\sqrt{n})$ quorum size and $O(1/\sqrt{n})$ load. Besides the already analyzed h-T-grid, all the other constructions have minimum quorum sizes $\approx \sqrt{2n}$. However, the h-triang is the only one where all quorums defined in the system have the same size. In the other two - Paths and Y -, quorums are obtained through paths in graphs, thus leading to a possible larger quorum size - for example, in system Y the authors indicate an average quorum size of ≈ 8.1 in a system with 28 elements [10] (against 7 in the h-triang). This larger average size imposes not only an increase in the number of messages necessary to obtain a quorum, but also a larger load - using a strategy that uniformly distributes the load with the above

	3x3 (9 nodes)		4x4 (16 nodes)		5x5 (25 nodes)		4x6 (24 nodes)	
p	h-grid	h-T-grid	h-grid	h-T-grid	h-grid	h-T-grid	h-grid	h-T-grid
0.1	0.016893	0.015213	0.005799	0.005361	0.001753	0.001621	0.001949	0.000611
0.2	0.109235	0.098585	0.069318	0.063866	0.039439	0.036300	0.034161	0.016690
0.3	0.286224	0.259783	0.243795	0.225066	0.191581	0.176290	0.167172	0.104402
0.5	0.716797	0.667969	0.746628	0.706604	0.751019	0.708871	0.725377	0.598435

Table 1. Failure probability in the hierarchical grid and hierarchical T-grid quorum systems.

p	Majority (15)	HQS (15)	CWlog (14)	h-T-grid (16)	Paths (13)	Y (15)	h-triang (15)
0.1	0.000034	0.000210	0.001639	0.015213	0.007351	0.000745	0.000677
0.2	0.004240	0.009567	0.021787	0.098585	0.063493	0.017603	0.016577
0.3	0.050013	0.070946	0.099915	0.259783	0.206296	0.093599	0.090712
0.5	0.500000	0.500000	0.500000	0.667969	0.662598	0.500000	0.500000

Table 2. Failure probability in quorum systems with approximately 15 nodes.

p	Majority (28)	HQS (27)	CWlog (29)	h-T-grid (25)	Paths (25)	Y (28)	h-triang (28)
0.1	0.000000	0.000016	0.000205	0.001621	0.001201	0.000057	0.000055
0.2	0.000229	0.002681	0.006865	0.036300	0.025045	0.005012	0.004851
0.3	0.014257	0.039626	0.056988	0.176290	0.136541	0.052777	0.051670
0.5	0.500000	0.500000	0.500000	0.708872	0.678858	0.500000	0.500000

Table 3. Failure probability in quorum systems with approximately 28 nodes.

Num. nodes		Majority	HQS	CWlog	h-T-grid	Paths	Y	h-triang
≈ 15	max.	8	6	6	7	-	6	5
	min.	8	6	3	4	5	5	5
	load	53.3%	40%	55.5%	41%($\geq$ 36.5%))	$\geq$ 39.2%	34.6%	33.3%
≈ 28	max.	14	8	10	9	-	11	7
	min.	14	8	4	5	7	7	7
	load	51%	29.6%	43.7%	34%($\geq$ 29.7%)	$\geq$ 28.2%	28.9%	25%
≈ 100	max.	51	≈ 19	25	19	-	-	14
	min.	51	≈ 19	5	10	15	14	14

Table 4. Minimum and maximum quorum sizes and load.

S	c(S)	same quorum size	$\mathcal{L}(\mathcal{S})$
Majority	$\frac{n+1}{2}$	Yes	$1/2$
HQS	$n^{0.63}$	Yes	$n^{-0.37}$
CWlog	$\lg n - \lg\lg n$	No	$1/\lg n$
h-T-grid	$\sqrt{n}$	No (avg. $> \frac{3}{2}\sqrt{n}$)	$> \frac{3}{2}\sqrt{n}$
Paths	$\approx \sqrt{2n}$	No	$\frac{\sqrt{2}}{\sqrt{n}} \leq \mathcal{L}(Paths) \leq \frac{2\sqrt{2}}{\sqrt{n}}$
Y	$\approx \sqrt{2n}$	No	$> \sqrt{2}/\sqrt{n}$
h-triang	$\approx \sqrt{2n}$	Yes	$\sqrt{2}/\sqrt{n}$

Table 5. Properties of quorum system constructions.

average quorum size in a system with 28 elements, the load of the system Y is 28.9% (against 25% in the h-triang). In table 4 we present the quorum sizes for systems with approximately 15, 28 and 100 elements. In table 5 we present the approximate asymptotic values of the minimum quorum size and load for the different systems. From the presented values it is possible to observe that the h-triang presents the best load (the strategy that induces such load is presented in the previous section). Moreover, from the systems that present $O(1/\sqrt{n})$ load it is the only one that has a fixed quorum size and presents the lower average quorum size.

7. Summary

In this paper, we have proposed a small modification to the hierarchical grid quorum system that reduces the size of the quorums used and improves the availability and load. From our analysis the failure probability can be further increased in the modified construction using slightly rectangular grids instead of square grids (the same situation does not occur in the original construction).

We have also proposed a new quorum system based on a hierarchical organization using a triangle shape. This new construction presents better availability and load than the grid-based constructions. The quorum size is constant and it is smaller than the average quorum size in those systems. The load is almost optimal ($\sqrt{2}/\sqrt{n}$), and it is the best from the analyzed systems that present high availability (the system proposed in [13] has optimal load - $1/\sqrt{n}$ - but poor asymptotic availability). It has a quorum size smaller than the average of the quorum size in the studied systems with $O(1/\sqrt{n})$ load and the availability is also the best for the analyzed number of elements in these systems.

References

[1] D. Agrawal, A. El-Abbadi. An efficient and fault-tolerant solution for distributed mutual exclusion. *ACM Transactions on Computer Systems*, February 1991.

[2] Y. Chang. A simulation study on distributed mutual exclusion. *Journal of Parallel and Distributed Computing*, vol. 33, 1996.

[3] S. Cheung, M. Ammar, M. Ahamad. The grid protocol: a high performance scheme for maintaining replicated data. In *Proceedings of the* 6^{th} *International Conference on Data Engineering*, February 1990.

[4] A. Fu, Y.S. Wong, M.H. Wong. Diamond Quorums Consensus for High Capacity and Efficiency in a Replicated Database System. *Distributed and Parallel Databases, An International Journal*, 8(4), 2000

[5] D. Gifford. Weighted voting for replicated data. In *Proceedings of the* 7^{th} *ACM Symposium on Operating Systems Principles*, December 1979.

[6] J. Holliday, R. Steinke, D. Agrawal, A. El-Abbadi. Epidemic Quorums for Managing Replicated Data. In *Proceedings of the* 19^{th} *IEEE Int. Performance, Computing, and Communications Conference (IPCCC 2000)*, February 2000.

[7] P. Keleher. Decentralized Replicated-Object Protocols. In *Proceedings of the* 18^{th} *Annual ACM Symposium on Principles of Distributed Computing*, April 1999.

[8] A. Kumar. Hierarchical Quorum Consensus: A New Algorithm for Managing Replicated Data. *IEEE Transactions on Computers*, September 1991.

[9] A. Kumar, S. Cheung. A high availability $\sqrt{n}$ hierarchical grid algorithm for replicated data. *Inf. Proc. Letters*, 40, 1991.

[10] Y. Kuo, S. Huang. A Geometric approach for Constructing Coteries and k-Coteries. *IEEE Transactions on Paralell and Distributed Systems*, April 1997.

[11] W. Luk, T. Wong. Two new quorum based algorithms for distributed mutual exclusion. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS'1997)*, 1997.

[12] D. Malkhi, M. Reiter, and A. Wool. The load and availability of Byzantine quorum systems. *SIAM Journal Computing*, 29(6), 2000.

[13] M. Maekawa. A $\sqrt{n}$ algorithm for mutual exclusion in decentralized system. *ACM Transactions on Computer Systems*, May 1985.

[14] M. Naor, A. Wool. The load, capacity and availability of quorum systems. *SIAM Jounal Computing*, April 1998.

[15] D. Peleg, A. Wool. The availability of quorum systems. *Information and Computation*, 123(2), 1995.

[16] D. Peleg, A. Wool. Crumbling walls: A class of practical and efficient quorum systems. *Distributed Computing*, 10(2):87-98, 1997.

[17] R. Prakash, M. Singhal. Dynamic Hashing + Quorum = Efficient Location Management for Mobile Computing Systems. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, 1997.

[18] A. Wool. Quorum systems in replicated databases: Science or fiction?. *Bulletin of the IEEE Technical Committee on Data Engineering*, December 1998.